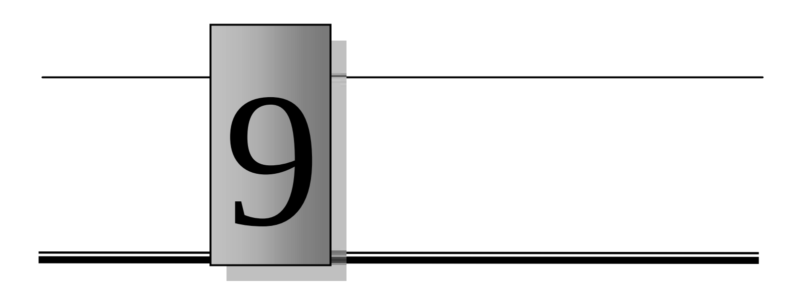
الباب
الرابع



الباب التاسع

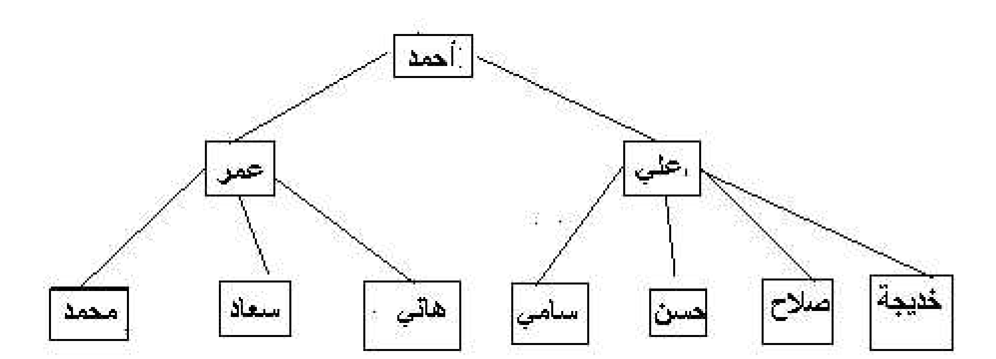
الأشجار Trees

Main Reference:
Discrete Mathematics and its Applications.
Kenneth H. Rosen

9.1 مقدمة

يستخدم مصطلح (الشجرة) في تراكيب البيانات لتمثيل بعض البيانات بطريقة مماثلة للشجرة الفعلية. فالشجرة لها جذر وأغصان وأوراق، وقد وجد الانسان منذ القدم أن أفضل طريقة لمتابعة أصول عائلته هي طريقة الشجرة، وسماها "شجرة

العائلة". وبعد اختراع الحاسوب وجد المبرمجون أن أفضل طريقة لتمثيل بعض البيانات داخل ذاكرة الحاسوب هي طريقة الشجرة.
لنأخذ مثلا شجرة عائلة كما في الشكل التالي:

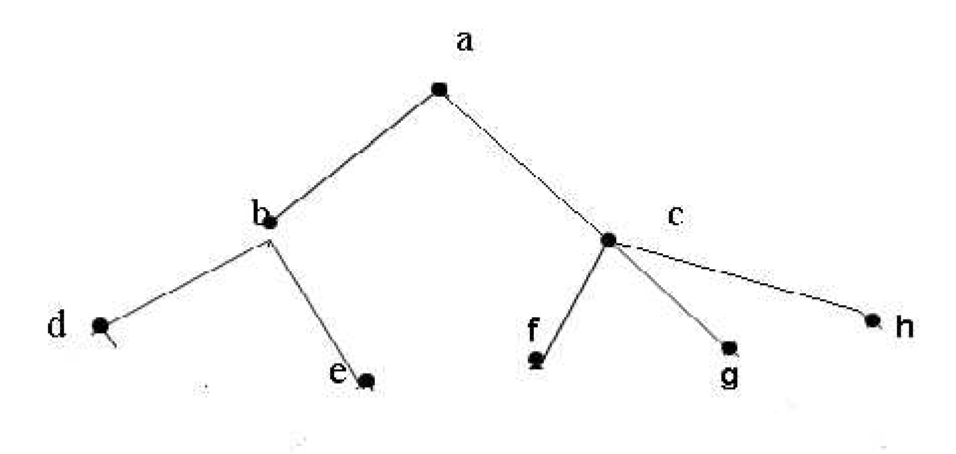


الشكل (9.1.1) شجرة عائلة

بمجرد النظر الى هذه الشجرة نستطيع فهم تركيبة هذه العائلة، فنلاحظ في هذه الشجرة أن اسم الجد هو "أحمد" وله ولدان هما "علي" و"عمر" وان "علي" له الأبناء " سامي وحسن وصلاح وخديجة" أما "عمر" فله من الأبناء " محمد وسعاد وهاني".
وكما نلاحظ فإن هذه الشجرة هي نوع خاص من الأشكال، وبالتحديد فهي عبارة عن ***شكل متصل connected graph وغير موجه undirected لا يوجد به مسار دائري***. تذكر أن المسار (الطريق) الدائري يكون فيه $V_0 = V_n$. أي أن آخر رأس vertex يتطابق مع أول رأس.

9.2 تعريفات

يبين الشكل 9.2.1 مثالا للشجرة.



الشكل (9.2.1) مثال لشجرة tree

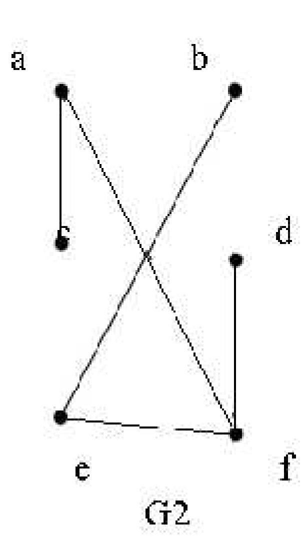
هذا الشكل يحقق الشرطين الأساسين في الشجرة : الاتصال وعدم وجود مسارات مغلقة. فمثلا يمكننا ايجاد مسار بسيط بين أي عقدتين (خاصية الاتصال) خذ مثلا المسار بين الرأس e والرأس f هو

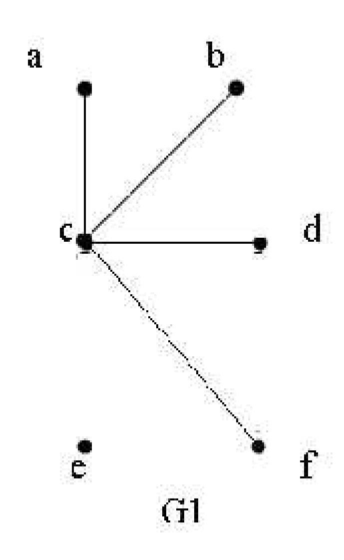
(e, b ,a , c, f)

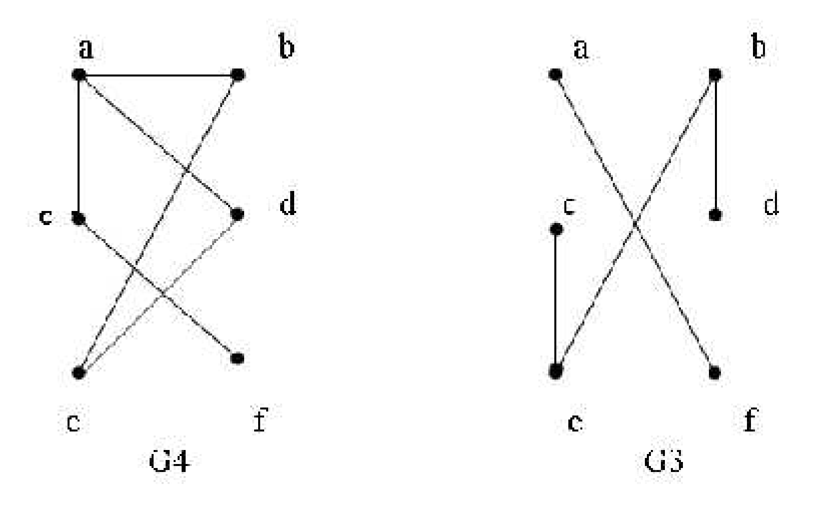
لو وضعنا حافة edge بين الرأس d والرأس e في هذا الشكل ، فلن يعتبر شجرة لوجود مسار مغلق هو (b, d, e) .

مثال 9.2.1

أي من الأشكال التالية يعتبر شجرة ؟







الشكل 9.2.2 أي من هذه الأشكال يعتبر شجرة؟

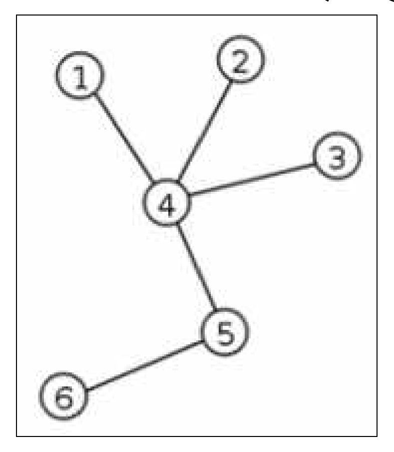
الإجابة: يعتبر الشكلان G1 , G2 من نوع الشجرة أما G3 , G4 فليس كذلك. والسبب أن G3 ليس شجرة هو كونه شكلا غير متصل فمثلا لايوجد مسار بين الراسين a , b، أما G4 فنجد فيه مسارا دائريا مثل:

(e, b, a, d, e).

الجذر (root) في الشجرة هو عبارة عن رأس يتجه منه مسار لكل رأس آخر في الشجرة. فمثلا في الشكل 9.2.1 يعتبر الرأس a جذرا لهذه الشجرة.

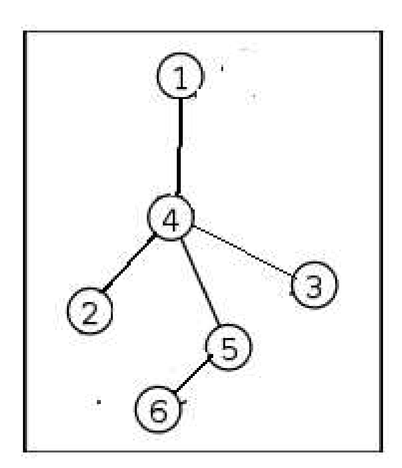
لاحظ أن الشجرة بدون جذر (unrooted) يمكن تحويلها إلى ذات جذر (rooted) وذلك باختيار رأس مناسب كجذر.

على سبيل المثال الشجرة التالية:



الشكل 9.2.3

التـي تحتـوي علـى 6 رؤوس vertices و 5 حـواف edges يمكـن اعـادة رسـمها بأخذ الرأس 1 كجذر على النحو التالي:



الشكل 9.2.4

وإذا كان v رأسا في الشجرة T (غير الجذر) فإن الرأس u يسمى **أب** (parent) للعقدة v إذا وجدت حافة موجهة من u إلى v. وفي هذه الحالة تسمى v **ابن** (child) للرأس u.



والرؤوس التي لها نفس الأب تسمى **إخوة** (siblings) . والرأس التي ليس لها أبناء تسمى **ورقة** (leaf).

والشجرة الفرعية subtree ذات الجذر a (حيث a رأس في الشجرة الأم) هي عبارة عن شكل فرعي subgraph يتكون من الرأس a وسلالته (descendent)

وأي رأس له أبناء children يسمى رأس **داخلي** internal vertex.

لاحظ أن:

عدد رؤوس الشجرة = عدد الرؤوس الداخلية + عدد الأوراق

مثال : ما هي الرؤوس الداخلية والأوراق في الشكل 9.25؟

الاجابة:

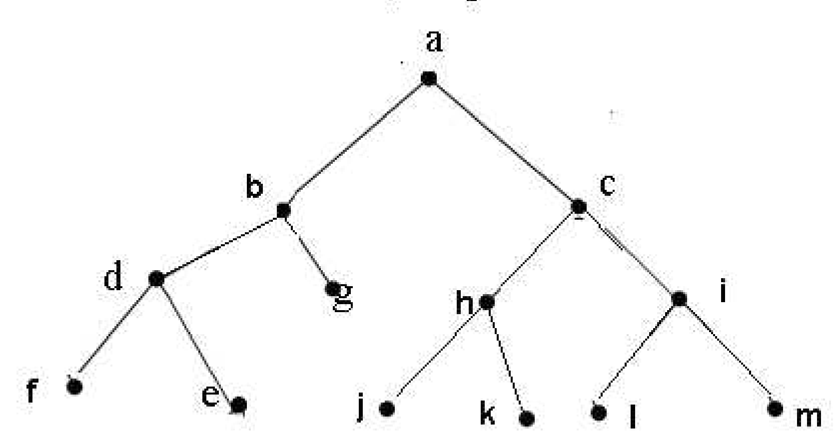
في هذا الشكل الرؤوس الداخلية هي:

a, b , c, d, g, h, i

أما الأوراق فهي:

f , e, j, k, l , m

<u>الشجرة الثنائية</u> binary tree هي الشجرة التي كل رأس داخلي فيها لا يزيد عدد أبنائه عن 2 ، كما في الشكل التالي:



الشكل 9.2.5 مثال لشجرة ثنائية

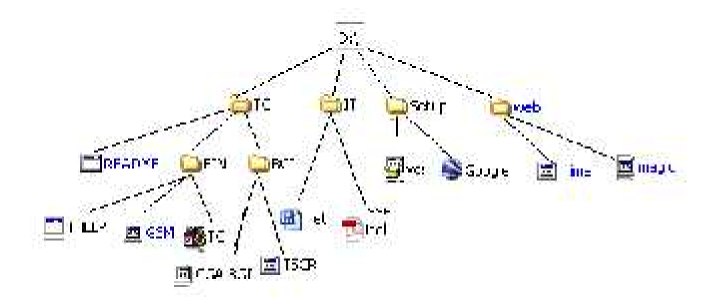
<u>ملاحظة :</u> بصورة عامة في الشجرة الثنائية يمكن أن يكون هناك رأس أو أكثر ذا ابن واحد، ولكن إذا كان عدد الأبناء في الشجرة لايزيد عن 2 فهي تسمى شجرة ثنائية أما إذا كان عدد الأبناء لأي رأس هو دائما 2 فتسمى ثنائية كاملة full binary tree.

9.3 أمثلة تطبيقية للأشجار

(1) شجرة الملفات في الحاسوب

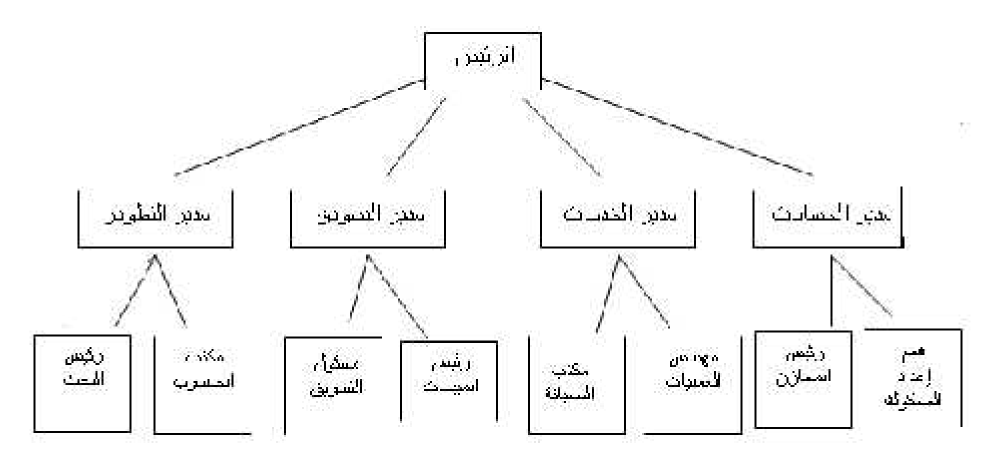
تقسم الملفات على القرص في مجلدات (أو أدلة) folders ، كل مجلد يحتوي على مجلدات أو ملفات. في هذه الحالة يسمى الجذر root directory ، والأدلة الفرعية subdirectories هي رؤوس داخلية، أما الملفات فهي عبارة عن أوراق leaves.

فمثلا قد يكون القرص :D متكونا من المجلدات والملفات التالية:



الشكل 9.3.1 مثال لشجرة الملفات في القرص :D

(2) التطبيق الثاني في الهيكلية الإدارية لشركة هو مثال شائع لتطبيق الشجرة :

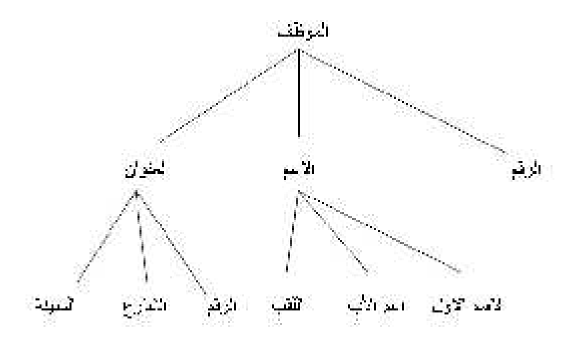


الشكل 9.3.2 هيكلية إدارية لشركة

قسم المراجعة

(3) سجل الموظف Employee record

يعتبر شجرة كما موضح في الشكل التالي:



الشكل 9.3.4 سجل موظف

التطبيق الثالث : تمثيل العمليات الحسابية

يمكن استعمال الأشجار في تمثيل العبارات الحسابية arithmetic expressions من حيث أولويات العمليات ، وذلك باعتبار العمليات كرؤوس في الشجرة ، واعتبار الأعداد والمتغيرات كأوراق الشجرة.

مثال1: ارسم شجرة للعملية التالية

(x + 5) / (x-4)

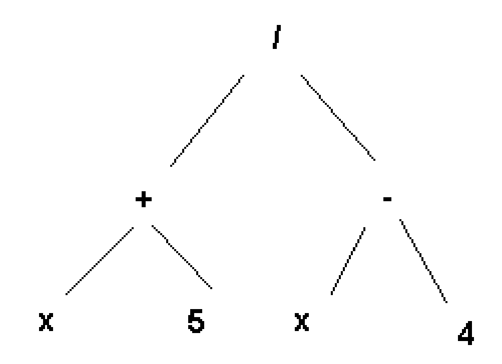
الاجابة : تتم العملية كما يلي:

1- الطرح x-4 لأن العملية بين قوسين

2- الجمع x+5 لأن العملية بين قوسين.

3- القسمة

ويمكن رسم شجرة لها بحيث تتم العمليات من اليسار إلى اليمين كما يلي:



الشكل 9.3.5 العملية (x + 5) / (x-4)

مثال 2 : ارسم شجرة للعملية التالية:

$x + y/x + 7$

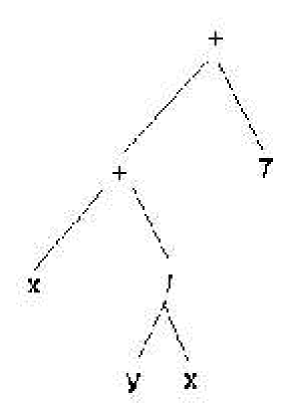
الاجابة: نلاحظ أن أولوية العمليات هنا كما يلي:

1- قسمة y على x

2- اضافة x الى y/x لأنها على اليسار

3- اضافة 7 الى الناتج من الخطوة 2.

يمكن تمثيل ذلك بالشجرة التالية:

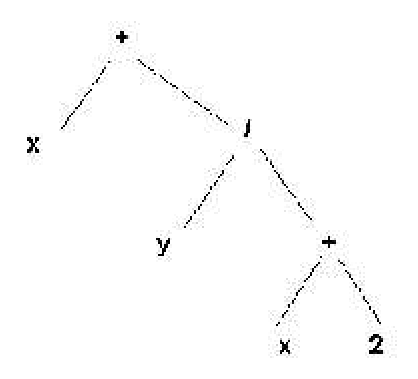


الشكل 9.3.6 العملية $x + y/x + 7$

لاحظ أن رؤوس الشجرة يتم تتبعها من اليسار الى اليمين.

مثال 3:

ما هي العملية التي يمثلها الشكل 9.3.7 ؟



الشكل 9.3.7 ما هي هذه العملية؟

الاجابة:

تقرأ الشجرة من اليسار الى اليمين فنحصل على

x + y/(x+2)

<u>**التطبيق الرابع**</u> : شجرة القرار Decision Tree

تساعدنا تركيبة الشجرة في اتخاذ القرار وذلك بمقارنة جميع الاحتمالات التي يمكن أن تحدث.

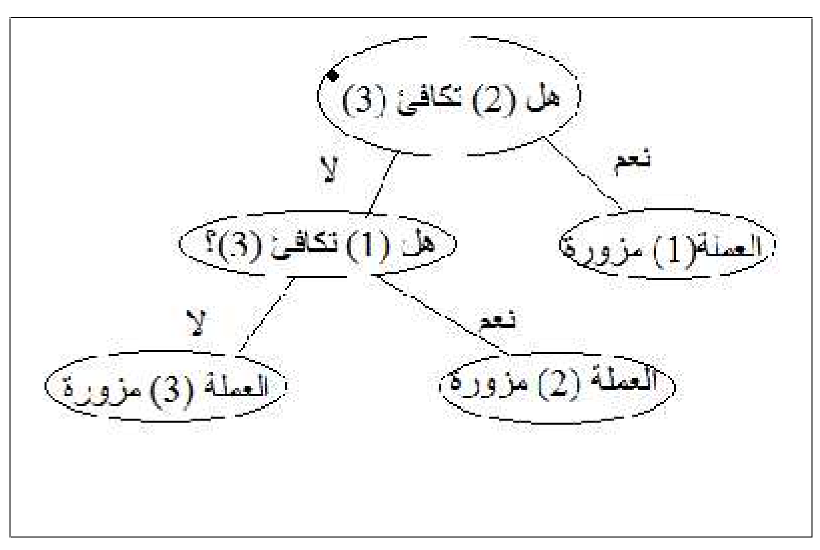
على سبيل المثال نفترض أن لدينا 3 عملات معدنية تبدو متكافئة ولكننا نعلم أن اثنين منها متكافئة ، والثالثة مزورة ، ونعرف ذلك من وزنها فهي أقل وزنا من باقي العملات. كيف يمكننا معرفة العملة المزورة إذا كان لدينا ميزان يقارن بين عملتين ونعرف بواسطته أيهما أقل وزنا؟

الحل:

نقوم بترقيم العملات الثلاثة من 1 الى 3. أي أن لدينا 3 احتمالات هي:

1. العملة رقم (1) مزورة. في هذه الحالة العملتان (2) و(3) متكافئتان.
2. العملة رقم (2) مزورة. في هذه الحالة العملتان (1) و(3) متكافئتان.
3. العملة رقم (3) مزورة. في هذه الحالة العملتان (1) و(2) متكافئتان.

هذه الخوارزمية يمكن تمثيلها بشجرة القرار التالية:



الشكل 9.3.8 شجرة القرار لاكتشاف العملة المزورة

9.4 نظريات

(1) الشجرة ذات n رأس بها $n - 1$ حافة .

(2) إذا كان كل رأس داخلي له m ابن ، وكان عدد الرؤوس الداخلية يساوي i فإن الشجرة بها $n = mi + 1$ رأس.

أما عدد الأوراق فهو

$$L = \frac{(m-1)\,n + 1}{m}$$

مثال 9.4.1 :

في شجرة الموظف المذكورة في الشكل 9.3.4 نجد أن

m=3 وأيضا i=3 أي أن عدد الرؤوس n=3*3+1 =10

كما نجد أن

$$L = ((3-1)10+1)/3 = 21/3 = 7$$

أي أن عدد الأوراق في هذه الشجرة هو 7 وبالتحديد هي

1. الرقم
2. الاسم الأول
3. اسم الأب
4. اللقب
5. رقم (المنزل)
6. الشارع
7. المدينة

تعريفات :

/ مستوى الرأس level of a vertex هو طول الطريق من الجذر إلى ذلك الرأس ، حيث مستوى الجذر = 0

ل ارتفاع height الشجرة هو أكبر مستوى فيها ، أي أن الارتفاع هو طول أطول مسار فيها من الجذر إلى أي رأس.

نظرية:

في الشجرة التي كل رأس داخلي فيها له m من الأبناء وارتفاعها h، يكون عدد الرؤوس فيها:

$$n = 1 + m + m^2 + \ldots + m^h$$
$$= (m^{h+1} – 1)/(m-1)$$

وعدد الأوراق

$$L = m^h$$

وعدد الرؤوس الداخلية هو :

$$i = n – L = (m^h – 1)/(m-1)$$

الاثبات: استخدم الاستنتاج الرياضي:

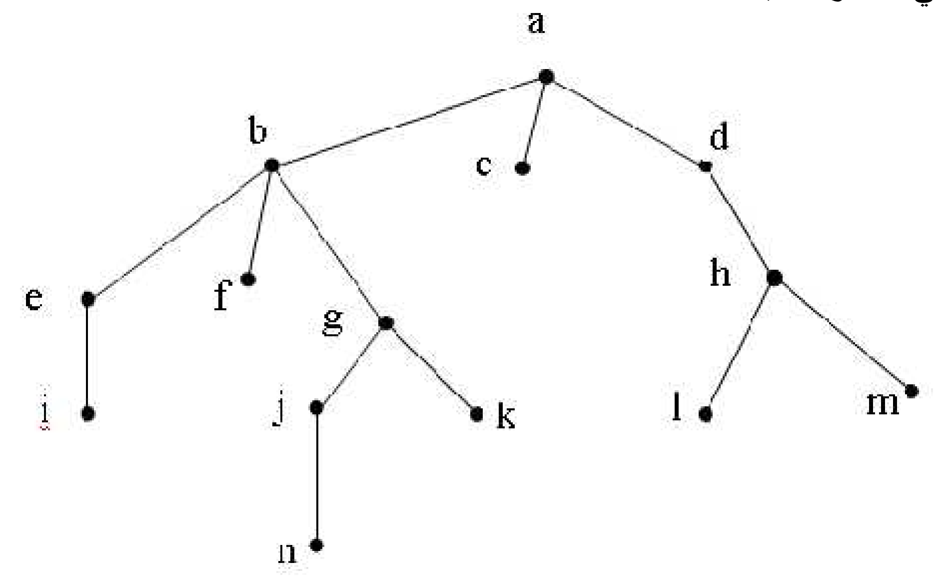
1. النظرية صحيحة في حالة h=1 لأن عدد الأبناء في هذه الحالة هو m^1 أي m .
2. الآن نفترض أن النظرية صحيحة عندما الارتفاع = h وبناء على ذلك نريد اثبات أن في حالة الارتفاع = h+1 فإن عدد الأوراق = $\mathbf{m^{h+1}}$. وهذا واضح لأن اضافة 1 للارتفاع يجعل كل ورقة من الأوراق التي عددها $\mathbf{m^h}$ رأسا داخليا له m من الأبناء . باستخدام قاعدة الضرب يصبح عدد الأوراق:

$$\mathbf{m^{h+1} = mm^h}$$

من (1) و(2) نرى أن النظرية صحيحة.

مثال 9.4.2

في الشجرة التالية :



الشكل 9.3.7 ما هو مستوى كل رأس في هذه الشجرة؟

مستوى a هو 0

مستوى b ، c ، d هو 1

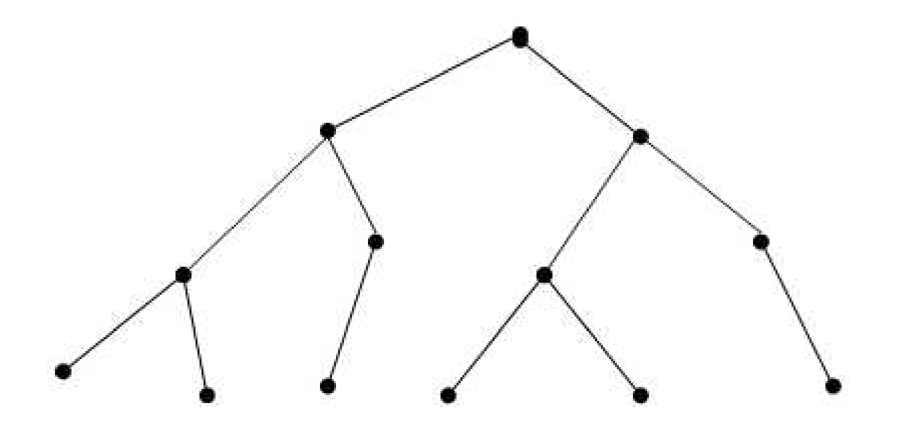
مستوى e ، f ، g ، h هو 2

مستوى i ، j ، k ، l ، m هو 3

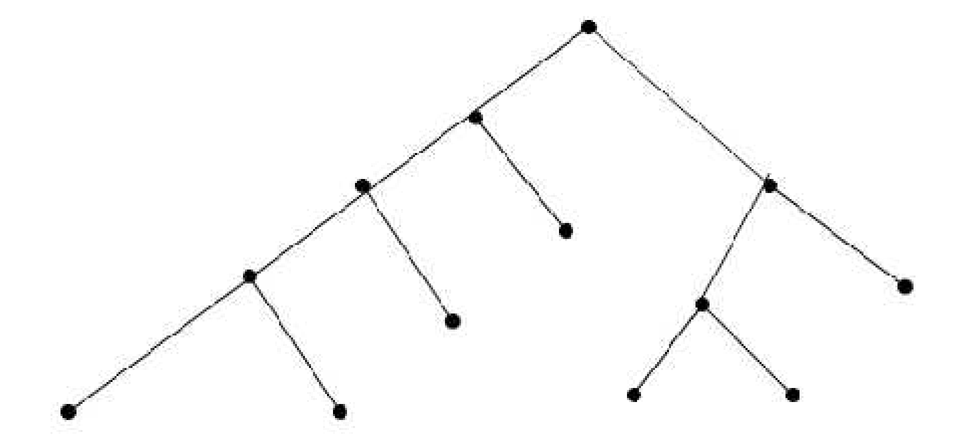
مستوى n هو 4

أي أن ارتفاع هذه الشجرة هو 4.

تعريـف : تسمى الشجرة متزنـة balanced إذا كـان مسـتوى كـل الأوراق هـو h أو $h-1$ حيث h هو ارتفاع الشجرة (height) .



الشكل 9.4.1. شجرة متزنة (ارتفاعها = 3)



الشكل 9.4.2 شجرة غير متزنة (يوجد بها مستويات 2 ، 3 ، 4 للأوراق)

9.5 تمثيل الشجرة في لغة باسكال

نبدأ أولاً ببعض التعريفات الأساسية :

1- الوحدة البيانية : (data item) وتعبر عن معلومة واحدة ، مثل الاسم أو العمر أو الدخل .

2- السجل (record) وهو مجموعة من الوحدات البيانية ذات علاقة. فالبيانات عن اسم الشخص مع عمره ودخله تعبر عن سجل يتكون من 3 وحدات بيانية (الاسم والعمر والدخل) .

3- الملف (file) وهو مجموعة من البيانات المتناظرة . كأن تكون سجلات عن طلبة كلية ما ، أو سجلات عن قطع الغيار المتوفر بالمخزن .

يمكن تمثيل الشجرة في لغة باسكال باستعمال السجل RECORD .

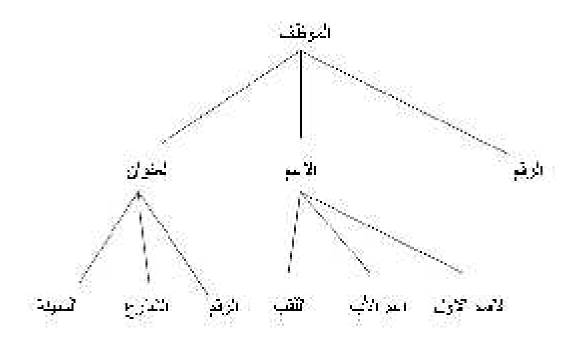
لتحديد نوع سجل و مكوناته ، نستخدم جملة TYPE على النحو :

```
TYPE r = RECORD
      f1: typef1;
      f2: typef2;
      .............
```

```
fn: typefn
END;
```

حيث r هو اسم السجل (الأب) (parent) ، و f1, f2, f3, … , fn هي أسماء مكونات السجل على الترتيب (الأبناء) (children) ، و typef1 هو نوع f1 ، و typef2 هو نوع f2 ، … الخ .

ملاحظة : من الممكن أن يكون الابن من النوع السجل RECORD كما في الشجرة التالية:



سجل الموظف employee في هذا الشكل يتكون من 3 حقول (رؤوس) هي:

1- الرقم num من نوع صحيح .
2- الاسم nm من نوع سجل .
3- العنوان adrs من نوع سجل.

```
TYPE employee=RECORD
    num : INTEGER;
    nm : name;
```

```
    adrs : address;
END;
```

على أن يتم تعريف السجل name والسجل address على النحو التالي:

```
TYPE name=RECORD
    first : STRING[20];
    middle : STRING[20];
    last : STRING[20];
END;
```

```
TYPE address=RECORD
    nmbr : STRING[20];
    street : STRING[20];
    city : STRING[20];
END;
```

فمثلا قراءة اسم الموظف تتم كما يلي:

```
read( emp.name.first)
read(emp.name.middle)
read(emp.name.last)
```

حيث يجب تعريف emp على أنه من نوع employee.

```
VAR emp : employee;
```

وبعد تحديد النوع employee ، يمكننا تحديد المتغير الذي يعبر عن ملف السجلات من النوع r (وليكن اسمه f) على النحو التالي :

VAR f: FILE OF employee;

مثال :

يبين البرنامج بالشكل (9.5.1) كيفية تكوين ملف نوعي (Typed file) . واسم هذا الملف (كما هو واضح في البرنامج) هو Studentf.dta

يبدأ هذا البرنامج بتحديد تركيبة السجل student في هذا الملف .

ويتحدد اسم السجل ومركباته في جملة TYPE . في هذا البرنامج ويتركب السجل (STUDENT) من 5 مكونات (fields) هي الرقم (number) و الاسم (name) و 3 درجات مختلفة (gr1, gr2, gr3) .

الشكل (9.5.1) : برنامج تكوين ملف نوعي على القرص

```
PROGRAM creatFile;
TYPE student=RECORD
     number : INTEGER;
     name : STRING[20];
     gr1, gr2, gr3 : REAL;
END;
VAR studentFile: FILE OF student;
    rec : student;
    i, num ,n : INTEGER;
    sname : STRING[20];
```

```
BEGIN
    ASSIGN(studentFile, 'studentf.dta');
    REWRITE(studentFile);
    WRITE('Enter number of students-->');
    READ(n);
    FOR i:= 1 TO n DO
    BEGIN
        WRITE('Enter number-->');
        READLN(rec.number);
        WRITE('Enter name-->');
        READLN(rec.name);
        WRITE('Enter grade1-->');
        READ(rec.gr1);
        WRITE('Enter second grade2-->');
        READ(rec.gr2);
        WRITE('Enter grade3-->');
        READ(rec.gr3);
        WRITE(studentFile,rec);
    END;
    CLOSE(studentFile);
END.
```

نلاحظ في البرنامج استخدام الجمل التالية :

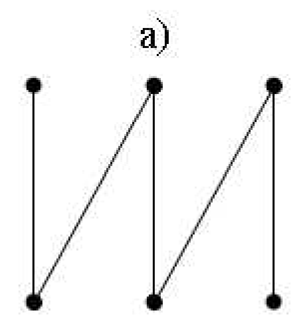
1- جملة ASSIGN لتعيين اسم الملف الذي يتم التخزين به على القرص .
2- جملة REWRITE لفتح ملف جديد للكتابة فيه (أي لتكوينه)
3- جملة WRITE لكتابة السجلات في الملف (وليس WRITELN).
4- جملة CLOSE لقفل الملف .

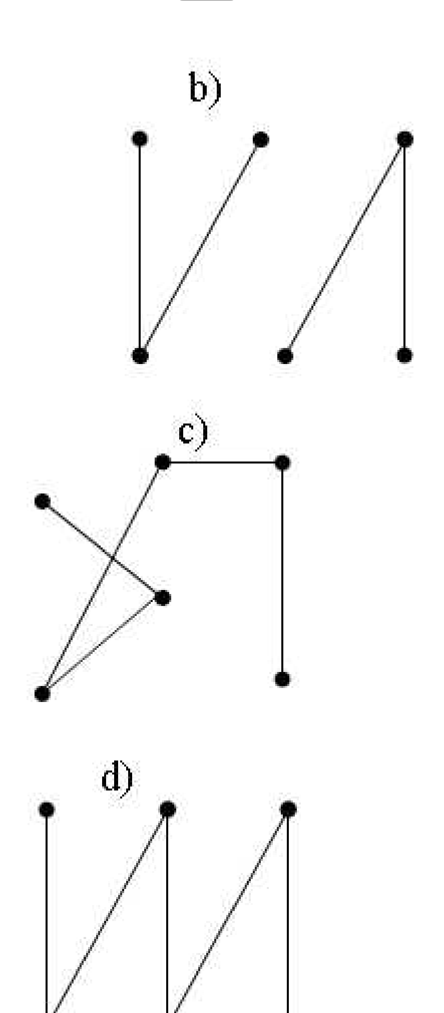
ونلاحظ أيضاً أن عدد السجلات التي تسجل في الملف طبقاً لهذا البرنامج هو 5 .

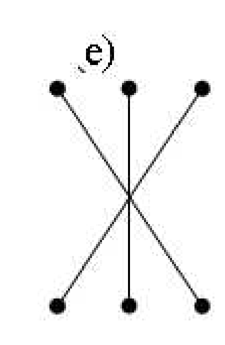
ملاحظة : إذا كان الملف موجوداً ، وثم فتح ملف باستخدام REWRITE فإن البيانات بالملف تضيع.

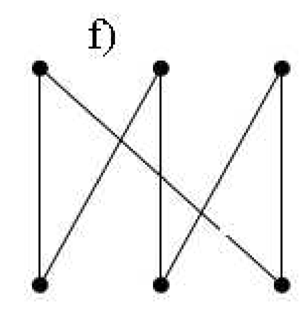
9.6 تمارين (19)

(1) أي من الأشكال التالية يعتبر شجرة ؟

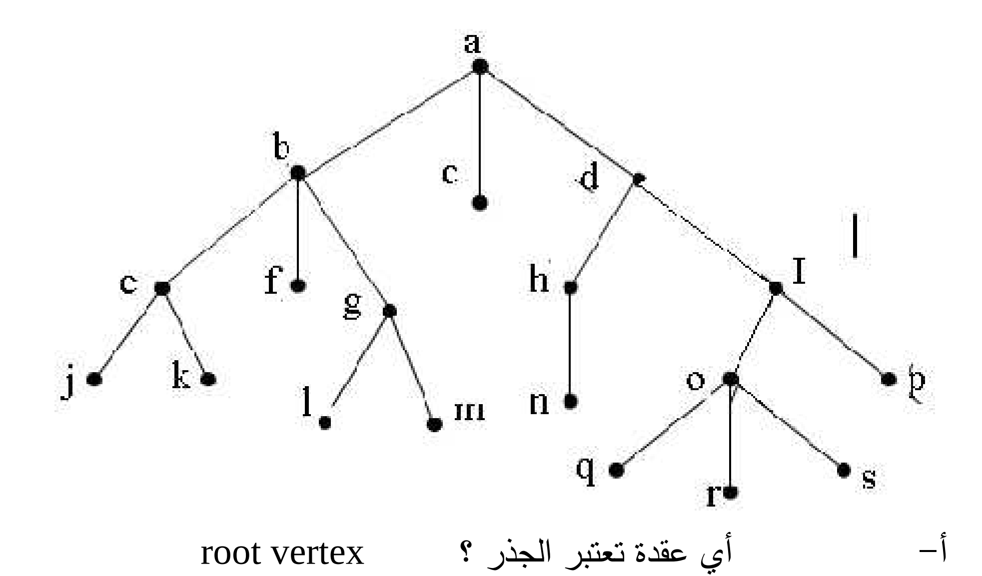



b)
c)
d)





(2) أجب عن الأسئلة التالية حول الشجرة المبينة :



أ- أي عقدة تعتبر الجذر ؟ root vertex

ب- أي عقد تعتبر داخلية؟ internal verities

ج- أي عقد تعتبر أوراق؟ leaves

د- أي عقد تعتبر أبناء I ؟ children

هـ- أي عقدة تعتبر أب h ؟ parent

و- أي عقد تعتبر أخوة o ؟ siblings

ز- أي عقد تعتبر سلالة b ؟ descendents

(3) في الشجرة في تمرين (2) :

أ- كم عدد أبناء كل عقدة داخلية ؟

ب- ما هو مستوى كل عقدة؟

ج- ما هو ارتفاع هذه الشجرة؟

د – هل تعتبر هذه الشجرة متزنة balanced ؟

(4) ارسم شجرة فرعية في تمرين (2) بحيث يكون جذرها عند :

أ– a

ب– c

ج– e

(5) ما هو ارتفاع شجرة ذات 85 رأس ، إذا كان كل رأس داخلي له 4 أبناء؟

(6) كم عدد الحواف في شجرة ثنائية كاملة بها 127 رأس داخلي ؟

(7) كم عدد الأوراق في شجرة بها 3280 رأسا داخليا إذا كان كل رأس داخلي به 3 أبناء؟

(8) ارسم الشجرة التي تمثل العملية الحسابية التالية:

ا– $(x + y) / (z + 9)$

ب– $5 + 7x - y$

ج – $(a – 6) /b + 8c + d$

(9) أرسم شجرة القرار لايجاد العملة المعدنية المزورة من بين 5 عملات (أربعة منها سليمة وواحدة مزورة وهي أقل وزنا من أي عملة أخرى) مستخدما ميزان يقارن بين الأوزان.